



The Functional Interpretation of Direct Computations

Ruy J.G.B. de Queiroz and Anjolina G. de Oliveira

Abstract

The concept of *direct computation* used by Statman (1977) was instrumental in the development of a notion of normal form for proofs of equality. In order to find a functional (Curry–Howard style) interpretation of direct computations we take a closer look at proof procedures for first-order sentences with equality drawing the attention to the need for introducing (function) symbols for rewrites. This leads us to a proposal to the effect that the framework of labelled natural deduction gives the right tools to formulate a proof theory for the “logical connective” of propositional equality in the style of the so-called Curry–Howard interpretation. The basic idea is that when analysing an equality sentence into (i) proof conditions (*introduction*) and (ii) immediate consequences (*elimination*), it becomes clear that we need to bring in identifiers (i.e. function symbols) for sequences of rewrites, and this is what we claim is the missing entity in P. Martin-Löf’s equality types (both intensional and extensional). For the proof system for equality we establish a normalisation procedure, proving that it is terminating and confluent.^a

^a Part of this material was presented in June 2006 at the *Logical Methods in the Humanities Seminar*, Stanford University, and the authors would like to thank Solomon Feferman and Grigori Mints for their comments and suggestions. The main results were produced while the first author was Edward Larocque Tinker Visiting Professor at the Department of Philosophy, Stanford Univ, and the second author was a Visiting Scholar at CSLI, Stanford.

Keywords: natural deduction, equality, labelled deduction, equality type.

1 Introduction

The clarification of the notion of normal form for equality reasoning took an important step with the work of Statman in the late 1970’s [35,36]. The concept of *direct computation* was instrumental in the development of Statman’s approach. By way of motivation, let us take a simple example from the λ -calculus.

$$\begin{aligned}
 &(\lambda x.(\lambda y.yx)(\lambda w.zw))v \triangleright_{\eta} (\lambda x.(\lambda y.yx)z)v \triangleright_{\beta} (\lambda y.yv)z \triangleright_{\beta} zv \\
 &(\lambda x.(\lambda y.yx)(\lambda w.zw))v \triangleright_{\beta} (\lambda x.(\lambda w.zw)x)v \triangleright_{\eta} (\lambda x.zx)v \triangleright_{\beta} zv \\
 &(\lambda x.(\lambda y.yx)(\lambda w.zw))v \triangleright_{\beta} (\lambda x.(\lambda w.zw)x)v \triangleright_{\beta} (\lambda w.zw)v \triangleright_{\eta} zv
 \end{aligned}$$

There is at least one sequence of conversions from the initial term to the final term. (In this case we have given three!) Thus, in the formal theory of λ -calculus, the

term $(\lambda x.(\lambda y.yx)(\lambda w.zw))v$ is declared to be **equal** to zv .

Now, some natural questions arise:

- (i) Are the sequences themselves *normal*?
- (ii) Are there non-normal sequences?
- (iii) If yes, how are the latter to be identified and (possibly) normalized?
- (iv) What happens if general rules of equality are involved?

Of course, if one considers only the β -contractions, the traditional choice is for the so-called *leftmost reduction* [8].

Nevertheless, we are interested in an approach to these questions that would be applicable both to λ -calculus and to proofs in Gentzen’s style Natural Deduction. As rightly pointed out by Le Chenadec in [5], the notion of normal proof has been somewhat neglected by the systems of equational logic: “In proof-theory, since the original work of Gentzen (1969) on sequent calculus, much work has been devoted to the normalization process of various logics, Prawitz (1965), Girard (1988). Such an analysis was lacking in equational logic (the only exceptions we are aware of are Statman (1977), Kreisel and Tait (1961)).” The works of Statman [35,36] and Le Chenadec [5] represent significant attempts to fill this gap. Statman studies proof transformations for the equational calculus E of Kreisel–Tait [11]. Le Chenadec defines an equational proof system (the *LE system*) and gives a normalization procedure.

What is a proof of an equality statement?

The so-called Brouwer–Heyting–Kolmogorov Interpretation defines logical connectives by taking proof, rather than truth-values, as a primitive notion:

a proof of the proposition:	is given by:
$A \wedge B$	a proof of A and a proof of B
$A \vee B$	a proof of A or a proof of B
$A \rightarrow B$	a function that turns a proof of A into a proof of B
$\forall x^D.P(x)$	a function that turns an element a into a proof of $P(a)$
$\exists x^D.P(x)$	an element a (witness) and a proof of $P(a)$

Based on the Curry–Howard functional interpretation of logical connectives, one can formulate the BHK-interpretation in formal terms as following:

a proof of the proposition:	has the canonical form of:
$A \wedge B$	$\langle p, q \rangle$ where p is a proof of A and q is a proof of B
$A \vee B$	$\text{inl}(p)$ where p is a proof of A or $\text{inr}(q)$ where q is a proof of B
	(‘inl’ and ‘inr’ abbreviate ‘into the left/right’)

$A \rightarrow B$	disjunct’)
$\forall x^D.P(x)$	$\lambda x.b(x)$ where $b(p)$ is a proof of B provided p is a proof of A
$\exists x^D.P(x)$	$\Lambda x.f(x)$ where $f(a)$ is a proof of $P(a)$ provided a is an arbitrary individual chosen from the domain D
	$\varepsilon x.(f(x), a)$ where a is a witness from the domain D , $f(a)$ is a proof of $P(a)$

(The term ‘ $\varepsilon x.(f(x), a)$ ’ is framed so as to formalise the notion of a function carrying its own argument [30].)

A question remains, however:

What is a proof of an equality statement?

An answer to such a question will help us extend the BHK-interpretation with an explanation of what is a proof of an equality statement:

a proof of the proposition: is given by:

$t_1 = t_2$?
	(Perhaps a sequence of rewrites starting from t_1 and ending in t_2 ?)

Two related questions naturally arise:

- (i) What is the logical status of the symbol “=”?
- (ii) What would be a canonical/direct proof of $t_1 = t_2$?

In a previous work [34] we have tried to show how the framework of labelled natural deduction can help us formulate a proof theory for the “logical connective” of propositional equality.¹ The connective is meant to be used in reasoning about equality between referents (i.e. the objects of the functional calculus), as well as with a general notion of substitution which is needed for the characterization of the so-called *term declaration logics* [2].

In order to account for the distinction between the equalities that are:

definitional, i.e. those equalities that are given as rewrite rules (equations),
or else originate from general functional principles (e.g. β , η , etc.),

and those that are:

propositional, i.e. the equalities that are supported (or otherwise) by an evidence (a composition of rewrites),

¹ An old question is in order here: what is a logical connective? We shall take it that from the point of view of proof theory (natural deduction style) a logical connective is whatever logical symbol which is analysable into rules of *introduction* and *elimination*.

we need to provide for an equality sign as a symbol for *rewrite* (i.e. as part of the functional calculus on the labels), and an equality sign as a symbol for a *relation* between referents (i.e. as part of the logical calculus on the formulas).

Definitional equalities.

Let us recall from the theory of λ -calculus, that:

Definition 1.1 [[8], (Def. 6.2 and Notation 7.1)] The formal theory of $\lambda\beta\eta$ equality has the following axioms:

$$\begin{aligned} (\alpha) \quad & \lambda x.M = \lambda y.[y/x]M & (y \notin FV(M)) \\ (\beta) \quad & (\lambda x.M)N = [N/x]M \\ (\eta) \quad & (\lambda x.Mx) = M & (x \notin FV(M)) \\ (\rho) \quad & M = M \end{aligned}$$

and the following inference rules:

$$\begin{aligned} (\mu) \quad & \frac{M = M'}{NM = NM'} & (\tau) \quad \frac{M = N \quad N = P}{M = P} \\ (\nu) \quad & \frac{M = M'}{MN = M'N} & (\sigma) \quad \frac{M = N}{N = M} \\ (\xi) \quad & \frac{M = M'}{\lambda x.M = \lambda x.M'} \\ (\zeta) \quad & \frac{Mx = Nx}{M = N} & \text{if } x \notin FV(MN) \end{aligned}$$

Propositional equality.

Again, let us recall from the theory of λ -calculus, that:

Definition 1.37 (β -equality) [8]

We say that P is β -equal or β -convertible to Q (notation $P =_\beta Q$) iff Q can be obtained from P by a finite (perhaps empty) series of β -contractions and reversed β -contractions and changes of bound variables. That is, $P =_\beta Q$ iff there exist P_0, \dots, P_n ($n \geq 0$) such that

$$\begin{aligned} (\forall i \geq n-1) & (P_i \triangleright_{1\beta} P_{i+1} \text{ or } P_{i+1} \triangleright_{1\beta} P_i \text{ or } P_i \equiv_\alpha P_{i+1}). \\ P_0 & \equiv P, \quad P_n \equiv Q. \end{aligned}$$

NB: equality with an **existential** force.

The same happens with $\lambda\beta\eta$ -equality:

Definition 7.5 ($\lambda\beta\eta$ -equality) [8]

The equality-relation determined by the theory $\lambda\beta\eta$ is called $=_{\beta\eta}$; that is, we define

$$M =_{\beta\eta} N \quad \Leftrightarrow \quad \lambda\beta\eta \vdash M = N.$$

Note again that two terms are $\lambda\beta\eta$ -equal if **there exists** a proof of their equality in the theory of $\lambda\beta\eta$ -equality.

Remark 1.2 In setting up a set of Gentzen’s ND-style rules for equality we need to account for:

1. definitional versus propositional equality;
2. there may be more than one *normal* proof of a certain equality statement;
3. given a (possibly non-normal) proof, the process of bringing it to a normal form should be finite and confluent.

The missing entity.

Within the framework of the functional interpretation (*à la* Curry–Howard [10]), the definitional equality is often considered by reference to a judgement of the form:

$$a = b : D$$

which says that a and b are equal elements from domain D . Notice that the ‘reason’ why they are equal does not play any part in the judgement. This aspect of ‘forgetting contextual information’ is, one might say, the first step towards ‘extensionality’ of equality, for whenever one wants to introduce intensionality into a logical system one invariably needs to introduce information of a ‘contextual’ nature, such as, where the identification of two terms (i.e. equation) comes from.

We feel that a first step towards finding an alternative formulation of the proof theory for propositional equality which takes care of the intensional aspect is to allow the ‘reason’ for the equality to play a more significant part in the form of judgement. We also believe that from the point of view of the logical calculus, if there is a ‘reason’ for two expressions to be considered equal, the proposition asserting their equality will be true, regardless of what particular composition of rewrites (definitional equalities) amounts to the evidence in support of the proposition concerned. Given these general guidelines, we shall provide what may be seen as a middle ground solution between the intensional [13,12] and the extensional [14] accounts of Martin-Löf’s propositional equality. The intensionality is taken care by the functional calculus on the labels, while the extensionality is catered by the logical calculus on the formulas. In order to account for the intensionality in the labels, we shall make the composition of rewrites (definitional equalities) appear as indexes of the equality sign in the judgement with a variable denoting a sequence of equality identifiers (we have seen that in the Curry–Howard functional interpretation there are at least four ‘natural’ equality identifiers: β , η , ξ and μ). So, instead of the form above, we shall have the following pattern for the equality judgement:

$$a =_s b : D$$

where ‘ s ’ is meant to be a sequence of equality identifiers.

In the sequel we shall be discussing in some detail the need to identify the kind of definitional equality, as well as the need to have a logical connective of ‘propositional equality’ in order to be able to reason about the functional objects (those to the left hand side of the ‘ $:$ ’ sign).

Term rewriting.

Deductive systems based on the Curry–Howard isomorphism [10] have an interesting feature: normalization and strong normalization (Church–Rosser property) theorems can be proved by reductions on the terms of the functional calculus. Exploring this important characteristic, we have proved these theorems for the *Labelled Natural Deduction – LND* [32,33] via a term rewriting system constructed from the *LND*-terms of the functional calculus [19]. Applying this same technique to the *LND* equational fragment, we obtain the normalization theorems for the equational logic of the *Labelled Natural Deduction System* [18,20,21].

This technique is used given the possibility of defining two measures of redundancy for the *LND* system that can be dealt with in the object level: the terms on the functional calculus and the *rewrite reason* (composition of rewrites), the latter being indexes of the equations in the *LND* equational fragment.

In the *LND* equational logic [29], the equations have the following pattern:

$$a =_s b : D$$

where one is to read: a is equal to b *because* of ‘ s ’ (‘ s ’ being the *rewrite reason*); ‘ s ’ is a term denoting a sequence of equality identifiers (β , η , α , etc.), i.e. a composition of rewrites. In other words, ‘ s ’ is the *computational path* from a to b .

In this way, the *rewrite reason* (reason, for short) represents an **orthogonal measure of redundancy** for the *LND*, which makes the *LND* equational fragment an “enriched” system of equational logic. Unlike the traditional equational logic systems, in *LND* equational fragment there is a gain in local control by the use of *reason*. All the proof steps are recorded in the composition of rewrites (reasons). Thus, consulting the reasons, one should be able to see whether the proof has the normal form. We have then used this powerful mechanism of controlling proofs to present a precise normalization procedure for the *LND* equational fragment. Since the reasons can be dealt with in the object level, we can employ a computational method to prove the normalization theorems: we built a term rewriting system based on an algebraic calculus on the “*rewrite reasons*”, which compute normal proofs. With this we believe we are making a step towards filling a gap in the literature on equational logic and on proof theory (natural deduction).

Kreisel–Tait’s system.

In [11] Kreisel and Tait define the system E for equality reasoning as consisting of axioms of the form $t = t$, and the following rules of inference:

$$\begin{array}{ll}
 (E1) & \frac{E[t/x] \quad t = u}{E[u/x]} \\
 (E2) & \frac{s(t) = s(u)}{t = u} \\
 (E3) & \frac{0 = s(t)}{A} \quad \text{for any formula } A \\
 (E4_n) & \frac{t = s^n(t)}{A} \quad \text{for any formula } A
 \end{array}$$

where t and u are terms, ‘0’ is the first natural number (zero), ‘ $s(-)$ ’ is the successor function.

Statman’s normal form theorem.

In order to prove the normalization results for the calculus E Statman defines two subsets of E : (i) a natural deduction based calculus for equality reasoning NE ; (ii) a sequent style calculus SE .

The NE calculus is defined as having axioms of the form $a = a$, and the rule of substituting equals for equals:

$$(=) \quad \frac{E[a/u] \quad a \approx b}{E[b/u]}$$

where E is any set of equations, and $a \approx b$ is ambiguously $a = b$ and $b = a$.

Statman arrives at various important results on normal forms and bounds for proof search in NE . In this case, however, a rather different notion of normal form is being used: the ‘cuts’ do not arise out of an *inversion principle*, as it is the case for the logical connectives, but rather from a certain form of sequence of equations which Statman calls *computation*, and whose normal form is called *direct computation*. With the formulation of a proof theory for the ‘logical connective’ of propositional equality we wish to analyse equality reasoning into its basic components: rewrites, on the one hand, and statements about the existence of rewrites, on the other hand. This type of analysis came to the surface in the context of constructive type theory and the Curry–Howard functional interpretation.

Martin–Löf’s Equality type.

There has been essentially two approaches to the problem of characterizing a proof theory for propositional equality, both of which originate in P. Martin–Löf’s work on *Intuitionistic Type Theory*: the intensional [13] and the extensional [14] formulations.

The extensional version.

In his [14] and [15] presentations of *Intuitionistic Type Theory* P. Martin-Löf defines the type of *extensional* propositional equality ‘ I ’ (here called ‘ I_{ext} ’) as:

I_{ext} -formation

$$\frac{D \text{ type} \quad a : D \quad b : D}{I_{ext}(D, a, b) \text{ type}}$$

I_{ext} -introduction

$$\frac{a = b : D}{\mathbf{r} : I_{ext}(D, a, b)}$$

I_{ext} -elimination²

$$\frac{c : I_{ext}(D, a, b)}{a = b : D}$$

I_{ext} -equality

$$\frac{c : I_{ext}(D, a, b)}{c = \mathbf{r} : I_{ext}(D, a, b)}$$

Note that the above account of propositional equality does not ‘keep track of all proof steps’: both in the I_{ext} -introduction and in the I_{ext} -elimination rules there is a considerable loss of information concerning the deduction steps. While in the I_{ext} -introduction rule the ‘ a ’ and the ‘ b ’ do not appear in the ‘trace’ (the label/term alongside the logical formula), the latter containing only the canonical element ‘ \mathbf{r} ’, in the rule of I_{ext} -elimination all the trace that might be recorded in the label ‘ c ’ simply disappears from label of the conclusion. If by ‘intensionality’ we understand a feature of a logical system which identifies as paramount the concern with issues of *context* and *provability*, then it is quite clear that any logical system containing I_{ext} -type can hardly be said to be ‘intensional’: as we have said above, neither its *introduction* rule nor its *elimination* rule carry the necessary *contextual* information from the premise to the conclusion.

And, indeed, the well-known statement of the extensionality of functions can be proved as a theorem of a logical system containing the I_{ext} -type such as Martin-Löf’s *Intuitionistic Type Theory* [15]. The statement says that if two functions return the same value in their common codomain when applied to each argument of their common domain (i.e. if they are equal pointwise), then they are said to be (extensionally) equal. Now, we can construct a derivation of the statement written in the formal language as:

$$\forall f, g^{A \rightarrow B}. (\forall x^A. I_{ext}(B, \text{APP}(f, x), \text{APP}(g, x))) \rightarrow I_{ext}(A \rightarrow B, f, g)$$

by using the rules of proof given for the I_{ext} , assuming we have the rules of proof given for the implication and the universal quantifier.

² The set of rules given in [14] contained the additional *elimination* rule:

$$\frac{c : I(D, a, b) \quad d : C(\mathbf{r}/z)}{J(c, d) : C(c/z)}$$

which may be seen as reminiscent of the previous *intensional* account of propositional equality [13].

The intensional version.

Another version of the propositional equality, which has its origins in Martin-Löf's early accounts of *Intuitionistic Type Theory* [12,13], and is apparently in the most recent, as yet unpublished, versions of type theory, is defined in [37] and [17]. In a section dedicated to the *intensional vs. extensional* debate, [37] (p.633) says that:

“Martin-Löf has returned to an intensional point of view, as in Martin-Löf (1975), that is to say, $t = t' \in A$ is understood as “ t and t' are definitionally equal”. As a consequence the rules for identity types have to be adapted.”

If we try to combine the existing accounts of the *intensional* equality type ‘ I ’ [13,37,17], here denoted ‘ I_{int} ’, the rules will look like:

I_{int}-formation

$$\frac{D \text{ type} \quad a : D \quad b : D}{I_{int}(D, a, b) \text{ type}}$$

I_{int}-introduction

$$\frac{a : D}{\mathbf{e}(a) : I_{int}(D, a, a)} \quad \frac{a = b : D}{\mathbf{e}(a) : I_{int}(D, a, b)}$$

I_{int}-elimination

$$\frac{a : D \quad b : D \quad c : I_{int}(D, a, b) \quad \frac{[x:D] \quad d(x) : C(x, x, \mathbf{e}(x))}{J(c, d) : C(a, b, c)} \quad \frac{[x:D, y:D, z:I_{int}(D, x, y)] \quad C(x, y, z) \text{ type}}{J(c, d) : C(a, b, c)}}{J(c, d) : C(a, b, c)}$$

I_{int}-equality

$$\frac{[x : D] \quad [x : D, y : D, z : I_{int}(D, x, y)] \quad a : D \quad d(x) : C(x, x, \mathbf{e}(x)) \quad C(x, y, z) \text{ type}}{J(\mathbf{e}(a), d(x)) = d(a/x) : C(a, a, \mathbf{e}(a))}$$

With slight differences in notation, the ‘adapted’ rules for identity type given in [37] and [17] resembles the one given in [13]. It is called *intensional* equality because there remains no direct connection between judgements like ‘ $a = b : D$ ’ and ‘ $s : I_{int}(D, a, b)$ ’.

A labelled proof theory for propositional equality.

Now, it seems that an alternative formulation of propositional equality within the functional interpretation, which will be a little more elaborate than the extensional I_{ext} -type, and simpler than the intensional I_{int} -type, could prove more convenient from the point of view of the ‘logical interpretation’. It seems that whereas in the former we have a considerable loss of information in the *I_{ext}-elimination*, in the latter we have an *I_{int}-elimination* too heavily loaded with (perhaps unnecessary) information. If, on the one hand, there is an *overexplicitation* of information in I_{int} , on the other hand, in I_{ext} we have a case of *underexplicitation*. With the formulation of a proof theory for equality via labelled natural deduction we wish to find a middle ground solution between those two extremes.

2 Labelled deduction

The functional interpretation of logical connectives via deductive systems which use some sort of labelling mechanism [15,6,7] can be seen as the basis for a general framework characterizing logics via a clear separation between a functional calculus on the *labels*, i.e. the referents (names of individuals, expressions denoting the record of proof steps used to arrive at a certain formula, names of ‘worlds’, etc.) and a logical calculus on the formulas. The key idea is to make these two dimensions as harmonious as possible, i.e. that the functional calculus on the labels matches the logical calculus on the formulas at least in the sense that to every abstraction on the variables of the functional calculus there corresponds a discharge of an assumption-formula of the logical calculus. One aspect of such interpretation which stirred much discussion in the literature of the past ten years or so, especially in connection with *Intuitionistic Type Theory* [15], was that of whether the logical connective of propositional equality ought to be dealt with ‘extensionally’ or ‘intensionally’. Here we attempt to formulate what appears to be a middle ground solution, in the sense that the intensional aspect is dealt with in the functional calculus on the labels, whereas the extensionality is kept to the logical calculus. We also intend to demonstrate that the connective of propositional equality (cf. Aczel’s [1] ‘ \doteq ’) needs to be dealt with in a similar manner to ‘Skolem-type’ connectives (such as disjunction and existential quantification), where notions like *hiding*, *choice* and *dependent variables* play crucial rôles.

2.1 Identifiers for (compositions of) equalities

In the functional interpretation, where a functional calculus on the labels go hand in hand with a logical calculus on the formulas, we have a classification of equalities, whose identifications are carried along as part of the deduction: either β -, η -, ξ -, μ - or α -equality will have been part of an expression labelling a formula containing ‘ \doteq ’. There one finds the key to the idea of ‘hiding’ in the *introduction* rule, and opening local (Skolem-type) assumptions in the *elimination* rule. (Recall that in the case of disjunction we also have alternatives: either into the left disjunct, or into the right disjunct.) So, we believe that it is not unreasonable to start off the formalization of propositional equality with the parallel to the disjunction and existential cases in mind. Only, the witness of the type of propositional equality are not the ‘ a ’s and ‘ b ’s of ‘ $a = b : D$ ’, but the actual (sequence of) equalities (β -, η -, ξ -, α -) that might have been used to arrive at the judgement ‘ $a =_s b : D$ ’ (meaning ‘ $a = b$ ’ because of ‘ s ’), ‘ s ’ being a sequence made up of β -, η -, ξ - and/or α -equalities, perhaps with some of the general equality rules of reflexivity, symmetry and transitivity. So, in the *introduction* rule of the type we need to form the canonical proof as if we were *hiding* the actual sequence. Also, in the rule of *elimination* we need to open a new local assumption introducing a new variable denoting a possible sequence as a (Skolem-type) new constant. That is, in order to eliminate the connective ‘ \doteq ’ (i.e. to deduce something from a proposition like ‘ $\doteq_D (a, b)$ ’), we start by choosing a new variable to denote the reason why the two terms are equal: ‘let t be an expression

(sequence of equalities) justifying the equality between the terms’. If we then arrive at an arbitrary formula ‘ C ’ labelled with an expression where the t still occurs free, then we can conclude that the same C can be obtained from the \doteq -formula regardless of the identity of the chosen t , meaning that the label alongside C in the conclusion will have been abstracted from the free occurrences of t .

Observe that now we are still able to ‘keep track’ of all proof steps (which does not happen with Martin-Löf’s I_{ext} -type) [14,15], and we have an easier formulation (as compared with Martin-Löf’s I_{int} -type) [13] of how to perform the *elimination* step.

2.2 The proof rules

In formulating the propositional equality connective, which we shall identify by ‘ \doteq ’, we shall keep the pattern of inference rules essentially the same as the one used for the other logical connectives (as in, e.g. [30]), and we shall provide an alternative presentation of propositional equality as follows:

\doteq -introduction

$$\frac{a =_s b : D}{s(a, b) : \doteq_D (a, b)}$$

\doteq -reduction

$$\frac{\frac{a =_s b : D}{s(a, b) : \doteq_D (a, b)} \doteq -intr \quad \frac{[a =_t b : D] \quad d(t) : C}{\text{REWR}(s(a, b), \acute{t}d(t)) : C} \doteq -elim}{\triangleright_\beta} \quad \frac{[a =_s b : D] \quad d(s/t) : C}{\triangleright_\beta}$$

\doteq -induction

$$\frac{e : \doteq_D (a, b) \quad \frac{[a =_t b : D] \quad t(a, b) : \doteq_D (a, b)}{\text{REWR}(e, \acute{t}t(a, b)) : \doteq_D (a, b)} \doteq -intr}{\text{REWR}(e, \acute{t}t(a, b)) : \doteq_D (a, b)} \doteq -elim \quad \triangleright_\eta \quad e : \doteq_D (a, b)$$

where ‘ \acute{t} ’ is an abstractor which binds the occurrences of the (new) variable ‘ t ’ introduced with the local assumption ‘ $[a =_t b : D]$ ’ as a kind of ‘Skolem’-type constant denoting the (presumed) ‘reason’ why ‘ a ’ was assumed to be equal to ‘ b ’. (Recall the Skolem-type procedures of introducing new local assumptions in order to allow for the elimination of logical connectives where the notion of ‘hiding’ is crucial, e.g. disjunction and existential quantifier – in [30].)

Now, having been defined as a ‘Skolem’-type connective, ‘ \doteq ’ needs to have a conversion stating the non-interference of the newly opened branch (the local assumption in the \doteq -elimination rule) with the main branch. Thus, we have:

\doteq -(permutative) reduction

$$\frac{\frac{e : \doteq_D (a, b) \quad d(t) : C}{\text{REWR}(e, \acute{t}d(t)) : C}}{w(\text{REWR}(e, \acute{t}d(t))) : W} \mathbf{r} \quad \triangleright_{\zeta} \quad \frac{\frac{e : \doteq_D (a, b) \quad d(t) : C}{w(d(t)) : W} \mathbf{r}}{\text{REWR}(e, \acute{t}w(d(t))) : W}$$

provided w does not disturb the existing dependencies in the term e (the main branch), i.e. provided that rule ‘ \mathbf{r} ’ does not discharge any assumption on which ‘ $\doteq_D (a, b)$ ’ depends. The corresponding ζ -equality is:

$$w(\text{REWR}(e, \acute{t}d(t))) =_{\zeta} \text{REWR}(e, \acute{t}w(d(t)))$$

The equality indicates that the operation w can be pushed inside the ‘-abstraction term, provided that it does not affect the dependencies of the term e .

Since we are defining the logical connective ‘ \doteq ’ as a connective which deals with singular terms, where the ‘witness’ is supposed to be hidden, we shall not be using direct *elimination* like Martin-Löf’s I_{ext} -*elimination*. Instead, we shall be using the following \doteq -*elimination*:

$$\frac{e : \doteq_D (a, b) \quad \frac{[a =_t b : D] \quad d(t) : C}{\text{REWR}(e, \acute{t}d(t)) : C}}{\text{REWR}(e, \acute{t}d(t)) : C}$$

The *elimination* rule involves the introduction of a new local assumption (and corresponding variable in the functional calculus), namely ‘ $[a =_t b : D]$ ’ (where ‘ t ’ is the new variable) which is only discharged (and ‘ t ’ bound) in the conclusion of the rule. The intuitive explanation would be given in the following lines. In order to eliminate the equality \doteq -connective, where one does not have access to the ‘reason’ (i.e. a sequence of ‘ β ’, ‘ η ’, ‘ ξ ’ or ‘ ζ ’ equalities) why the equality holds because ‘ \doteq ’ is supposed to be a connective dealing with singular terms (as are ‘ \vee ’ and ‘ \exists ’), in the first step one has to open a new local assumption supposing the equality holds because of, say ‘ t ’ (a new variable). The new assumption then stands for ‘let t be the unknown equality’. If a third (arbitrary) statement can be obtained from this new local assumption via an unspecified number of steps which does not involve any binding of the new variable ‘ t ’, then one discharges the newly introduced local assumption binding the free occurrences of the new variable in the label alongside the statement obtained, and concludes that that statement is to be labelled by the term ‘ $\text{REWR}(e, \acute{t}d(t))$ ’ where the new variable (i.e. t) is bound by the ‘-abstraction.

Another feature of the \doteq -connective which is worth noticing at this stage is the equality under ‘ ξ ’ of all its elements (see second *introduction* rule). This does not mean that the labels serving as evidences for the \doteq -statement are all identical to a constant (cf. constant ‘ \mathbf{r} ’ in Martin-Löf’s I_{ext} -type), but simply that if two (sequences of) equality are obtained as witnesses of the equality between, say ‘ a ’ and ‘ b ’ of domain D , then they are taken to be equal under ξ -equality. It would not seem unreasonable to think of the \doteq -connective of propositional equality as

expressing the proposition which, whenever true, indicates that the two elements of the domain concerned are equal under some (unspecified, *hidden*) composition of definitional equalities. It is as if the proposition points to the existence of a term (witness) which depends on both elements and on the kind of equality judgements used to arrive at its proof. So, in the logical side, one forgets about what was the actual witness. Cf. the existential generalization:

$$\frac{F(t)}{\exists x.F(x)}$$

where the actual witness is in fact ‘abandoned’. Obviously, as we are interested in keeping track of relevant information introduced by each proof step, in labelled natural deduction system the witness is not abandoned, but is carried over as an unbounded name in the label of the corresponding conclusion formula.

$$\frac{t : D \quad f(t) : F(t)}{\varepsilon x.(f(x), t) : \exists x^D.F(x)}$$

Note, however, that it is carried along *only* in the functional side, the logical side not keeping any trace of it at all.

Now, notice that if the functional calculus on the labels is to match the logical calculus on the formulas, than we must have the resulting label on the left of the ‘ \triangleright_β ’ as β -convertible to the concluding label on the right. So, we must have the convertibility equality:

$$\text{REWR}(s(a, b), \dot{t}d(t)) =_\beta d(s/t) : C$$

The same holds for the η -equality:

$$\text{REWR}(e, \dot{t}t(a, b)) =_\eta e : \dot{=}^D (a, b)$$

Parallel to the case of disjunction, where two different **constructors** distinguish the two alternatives, namely ‘**inl**’ and ‘**inr**’, we here have any (sequence of) equality identifiers (‘ β ’, ‘ η ’, ‘ μ ’, ‘ ξ ’, etc.) as **constructors** of proofs for the $\dot{=}$ -connective. They are meant to denote the alternatives available.

General rules of equality.

Apart from the already mentioned ‘constants’ (identifiers) which compose the reasons for equality (i.e. the indexes to the equality on the functional calculus), it is reasonable to expect that the following rules are taken for granted: *reflexivity*, *symmetry* and *transitivity*.

Substitution without involving quantifiers.

We know from logic programming, i.e. from the theory of unification, that substitution can take place even when no quantifier is involved. This is justified when,

for some reason a certain referent can replace another under some condition for identifying the one with the other.

Now, what would be counterpart to such a ‘quantifier-less’ notion of substitution in a labelled natural deduction system. Without the appropriate means of handling equality (definitional and propositional) we would hardly be capable of finding such a counterpart. Having said all that, let us think of what we ought to do at a certain stage in a proof (deduction) where the following two premises would be at hand:

$$a =_g y : D \quad \text{and} \quad f(a) : P(a)$$

We have that a and y are equal (‘identifiable’) under some arbitrary sequence of equalities (rewrites) which we name g . We also have that the predicate formula $P(a)$ is labelled by a certain functional expression f which depends on a . Clearly, if a and y are ‘identifiable’, we would like to infer that P , being true of a , will also be true of y . So, we shall be happy in inferring (on the logical calculus) the formula $P(y)$. Now, given that we ought to compose the label of the conclusion out of a composition of the labels of the premises, what label should we insert alongside $P(y)$? Perhaps various good answers could be given here, but we shall choose one which is in line with our ‘keeping record of what (relevant) data was used in a deduction’. We have already stated how much importance we attach to names of individuals, names of formula instances, and of course, what kind of deduction was performed (i.e. what kind of connective was introduced or eliminated). In this section we have also insisted on the importance of, not only ‘classifying’ the equalities, but also having variables for the kinds of equalities that may be used in a deduction. Let us then formulate our rule of ‘quantifier-less’ substitution as:

$$\frac{a =_g y : D \quad f(a) : P(a)}{g(a, y) \cdot f(a) : P(y)}$$

which could be explained in words as follows: if a and y are ‘identifiable’ due to a certain g , and $f(a)$ is the evidence for $P(a)$, then let the composition of $g(a, y)$ (the label for the propositional equality between a and y) with $f(a)$ (the evidence for $P(a)$) be the evidence for $P(y)$.

By having this extra rule of substitution added to the system of rules of inference, we are able to validate one half of the so-called ‘Leibniz’s law’, namely:

$$\forall x^D \forall y^D. (\dot{=}^D (x, y) \rightarrow (P(x) \rightarrow P(y)))$$

The *LND* equational fragment.

As we already mentioned, in the *LND* equational logic, the equations have an index (the *reason*) which keeps all proof steps. The *reasons* is defined by the kind of rule used in the proof and the equational axioms (*definitional equalities*) of the system. The rules are divided into the following classes: (i) general rules; (ii) subterm substitution rule; (iii) ξ - and μ -rules.

Since the *LND* system is based on the Curry–Howard isomorphism [10], terms represent proof constructions, thus proof transformations correspond to equalities

between terms. In this way, the *LND* equational logic can deal with equalities between *LND* proofs. The proofs in the *LND* equational fragment which deals with equalities between deductions are built from the basic proof transformations for the *LND* system, given in [32,30,33]. These basic proof transformations form an equational system, composed by *definitional equalities* (β , η and ζ).

General rules.

Definition 2.1 (equation) An equation in LND_{EQ} is of the form:

$$s =_r t : D$$

where s and t are terms, r is the identifier for the *rewrite reason*, and D is the type (formula).

Definition 2.2 (system of equations) A system of equations S is a set of equations:

$$\{s_1 =_{r_1} t_1 : D_1, \dots, s_n =_{r_n} t_n : D_n\}$$

where r_i is the *rewrite reason* identifier for the i th equation in S .

Definition 2.3 (rewrite reason) Given a system of equations S and an equation $s =_r t : D$, if $S \vdash s =_r t : D$, i.e. there is a deduction/computation of the equation starting from the equations in S , then the *rewrite reason* r is built up from:

- (i) the constants for *rewrite reasons*: $\{\rho, \beta, \eta, \zeta\}$;
- (ii) the r_i 's;

using the substitution operations:

- (iii) sub_L ;
- (iv) sub_R ;

and the operations for building new *rewrite reasons*:

- (v) σ, τ, ξ, μ .

Definition 2.4 (general rules of equality) The general rules for equality (reflexivity, symmetry and transitivity) are defined as follows:

$$\begin{array}{ccc} \text{reflexivity} & \text{symmetry} & \text{transitivity} \\ \frac{x : D}{x =_\rho x : D} & \frac{x =_t y : D}{y =_{\sigma(t)} x : D} & \frac{x =_t y : D \quad y =_u z : D}{x =_{\tau(t,u)} z : D} \end{array}$$

The “subterm substitution” rule.

Equational logic as usually presented has the following inference rule of substitution:

$$\frac{s = t}{s\theta = t\theta}$$

where θ is a substitution.

Note that the substitution θ “appeared” in the conclusion of the rule. As rightly pointed out by Le Chenadec in [5], from the view point of the subformula property (objects in the conclusion of some inference should be subobjects of the premises), this rule is unsatisfactory. He then defines two rules:

$$IL \frac{M = N \quad C[N] = O}{C[M] = O} \quad IR \frac{M = C[N] \quad N = O}{M = C[O]}$$

where M , N and O are terms and the context $C[]$ is adopted in order to distinguish subterms.

In [19] we have formulated an inference rule called “subterm substitution” which deals in a *explicit way*³ with substitutions. In fact, the *LND* [32,33] can be seen as an enriched system which brings to the object language terms, and now substitutions.

Definition 2.5 (subterm substitution) The rule of “subterm substitution” is framed as follows:

$$\frac{x =_r C \mid y \mid : D \quad y =_s u : D'}{x =_{\text{sub}_L(r,s)} C \mid u \mid : D} \quad \frac{x =_r w : D' \quad C \mid w \mid =_s u : D}{C \mid x \mid =_{\text{sub}_R(r,s)} u : D}$$

where C is the context in which the subterm detached by ‘ \mid ’ appears and D' could be a subdomain of D , equal to D or disjoint to D .

The symbols sub_L and sub_R denote in which side (L – *left* or R – *right*) is the premiss that contains the subterm to be substituted.

Note that the transitivity rule previously defined can be seen as a special case for this rule when $D' = D$ and the context C is empty.

³ In [32] de Queiroz and Gabbay recall Girard, who describes the intimate connections between constructivity and explicitation, and claim that “...one of the aims of inserting a label alongside formulas (accounting for the steps made to arrive at each particular point in the deduction) is exactly that of making *explicit* the use of formulas (and instances of formulas and individuals) throughout a deduction ...”

The ξ - and μ -rules.

In the Curry–Howard “formulae-as-types” interpretation [10], the ξ -rule⁴ states when two *canonical* elements are equal, and the μ -rule⁵ states when two *non-canonical* elements are equal. So, each introduction rule for the *LND* system has associated to it a ξ -rule and each elimination rule has a related μ -rule. For instance, the ξ -rule and μ -rule for the connective \wedge are defined as follows:

$$\frac{x =_u y : A \quad s =_v t : B}{\langle x, s \rangle =_{\xi(u,v)} \langle y, t \rangle : A \wedge B}$$

$$\frac{x =_r y : A \wedge B}{\text{FST}(x) =_{\mu(r)} \text{FST}(y) : A} \quad \frac{x =_r y : A \wedge B}{\text{SND}(x) =_{\mu(r)} \text{SND}(y) : B}$$

Term rewriting system for *LND* with equality.

In [20] we have proved termination and confluence for the rewriting system arising out of the proof rules given for the proposed natural deduction system for equality.

The idea is to analyse all possible occurrences of redundancies in proofs which involve the rules of rewriting, and the most obvious case is the nested application of the rule of symmetry. But there are a number of cases when the application of rewriting rules is redundant, but which is not immediately obvious that there is a redundancy. Take, for instance, the following case:

Definition 2.6 [reductions involving τ]

$$\frac{x =_r y : D \quad y =_{\sigma(r)} x : D}{x =_{\tau(r,\sigma(r))} x : D} \triangleright x =_{\rho} x : D$$

$$\frac{y =_{\sigma(r)} x : D \quad x =_r y : D}{y =_{\tau(\sigma(r),r)} y : D} \triangleright y =_{\rho} y : D$$

$$\frac{u =_r v : D \quad v =_{\rho} v : D}{u =_{\tau(r,\rho)} v : D} \triangleright u =_r v : D$$

⁴ The ξ -rule is the formal counterpart to Bishop’s constructive principle of definition of a set [4] (page 2) which says: “To define a set we prescribe, at least implicitly, what we have (the constructing intelligence) must do in order to construct an element of the set, and what we must do to show that two elements of the set are equal.” Cf. also [4] (page 12) Bishop defines a product of set as “The *cartesian product*, or simply *product*, $X \equiv X_1 \times \dots \times X_n$ of sets X_1, X_2, \dots, X_n is defined to be the set of all ordered n-tuples (x_1, \dots, x_n) and (y_1, \dots, y_n) of X are equal if the coordinates x_i and y_i are equal for each i .” See also [15] (p.8): “... a set A is defined by prescribing how a canonical element of A is formed as well as how two equal canonical elements of A are formed.” We also know from the theory of Lambda Calculus the definition of ξ -rule, see e.g. [3] (pp. 23 and 78): “ $\xi : M = N \Rightarrow \lambda x. M = \lambda x. N$ ”

⁵ The μ -rule is also defined in the theory of Lambda Calculus, see e.g. [16]: “The equational axioms and inference rules are as follows, where $[N/x]M$ denotes substitution of N for x in M

$$(\mu) \quad \frac{\Gamma \triangleright M_1 = M_2 : \sigma \Rightarrow \tau \quad \Gamma \triangleright N_1 = N_2 : \sigma}{\Gamma \triangleright M_1 N_1 = M_2 N_2 : \tau}$$

and is divided into two equalities μ and ν in [8] (p.66):

$$(\mu) \quad \frac{M = M'}{NM = NM'} \quad (\nu) \quad \frac{M = M'}{MN = M'N}$$

$$\frac{u =_{\rho} u : D \quad u =_r v : D}{u =_{\tau(\rho, r)} v : D} \triangleright u =_r v : D$$

Associated rewriting rule over the reason:

$$\begin{aligned} \tau(r, \sigma(r)) &\triangleright \rho \\ \tau(\sigma(r), r) &\triangleright \rho \\ \tau(r, \rho) &\triangleright r \\ \tau(\rho, r) &\triangleright r. \end{aligned}$$

Below is another less obvious case of occurrence of redundancy:

Definition 2.7 [$\beta_{rewr} \rightarrow$ -reduction]

$$\frac{\begin{array}{c} \vdots \\ b(x) =_r g(x) : B \end{array} \quad \frac{a =_s a' : A \quad \lambda x. b(x) =_{\xi(r)} \lambda x. g(x) : A \rightarrow B}{\text{APP}(\lambda x. b(x), a) =_{\mu(s, \xi(r))} \text{APP}(\lambda x. g(x), a') : B} \rightarrow \text{-intr}}{\text{APP}(\lambda x. b(x), a) =_{\mu(s, \xi(r))} \text{APP}(\lambda x. g(x), a') : B} \rightarrow \text{-elim} \triangleright b(a/x) =_r g(a'/x) : B$$

Associated rewriting rule:

$$\mu(s, \xi(r)) \triangleright r.$$

As an example:

Example 2.8

$$\frac{\frac{x =_r y : A}{\text{inl}(r) =_{\xi(r)} \text{inl}(y) : A \vee B} \quad \frac{\frac{x =_r y : A}{y =_{\sigma(r)} x : A}}{\text{inl}(y) =_{\xi(\sigma(r))} \text{inl}(x) : A \vee B}}{\text{inl}(x) =_{\tau(\xi(r), \xi(\sigma(r)))} \text{inl}(x) : A \vee B} \triangleright \frac{x : A}{x =_{\rho} x : A} \text{inl}(x) =_{\xi(\rho)} \text{inl}(x) : A \vee B$$

Associated rewriting rule over the reason:

$$\tau(\xi(r), \xi(\sigma(r))) \triangleright \xi(\rho).$$

By analysing all cases of redundant proofs in first-order logic with equality we arrive at following set of associated rewriting rules:

Definition 2.9 [$LND_{EQ}\text{-TRS}$]

1. $\sigma(\rho) \triangleright \rho$
2. $\sigma(\sigma(r)) \triangleright r$
3. $\tau(\mathcal{C}[r], \mathcal{C}[\sigma(r)]) \triangleright \mathcal{C}[\rho]$
4. $\tau(\mathcal{C}[\sigma(r)], \mathcal{C}[r]) \triangleright \mathcal{C}[\rho]$
5. $\tau(\mathcal{C}[r], \mathcal{C}[\rho]) \triangleright \mathcal{C}[r]$
6. $\tau(\mathcal{C}[\rho], \mathcal{C}[r]) \triangleright \mathcal{C}[r]$

7. $\text{sub}_L(\mathcal{C}[r], \mathcal{C}[\rho]) \triangleright \mathcal{C}[r]$
8. $\text{sub}_R(\mathcal{C}[\rho], \mathcal{C}[r]) \triangleright \mathcal{C}[r]$
9. $\text{sub}_L(\text{sub}_L(s, \mathcal{C}[r]), \mathcal{C}[\sigma(r)]) \triangleright s$
10. $\text{sub}_L(\text{sub}_L(s, \mathcal{C}[\sigma(r)]), \mathcal{C}[r]) \triangleright s$
11. $\text{sub}_R(\mathcal{C}[s], \text{sub}_R(\mathcal{C}[\sigma(s)], r)) \triangleright r$
12. $\text{sub}_R(\mathcal{C}[\sigma(s)], \text{sub}_R(\mathcal{C}[s], r)) \triangleright r$
13. (FST) $\mu(\xi(r, s)) \triangleright r$
14. (SND) $\mu(\xi(r, s)) \triangleright s$
15. (inl) $\mu(\xi(r), s, u) \triangleright s$
16. (inr) $\mu(\xi(r), s, u) \triangleright u$
17. ($\beta_{rewr} - \{\rightarrow, \forall\}$) $\mu(s, \xi(r)) \triangleright r$
18. ($\beta_{rewr} - \exists$) $\mu(\xi(r), s) \triangleright s$
19. ($\eta_{rewr} - \wedge$) $\xi(\mu(r)) \triangleright r$
20. ($\eta_{rewr} - \vee$) $\mu(t, \xi(r), \xi(s)) \triangleright t$
21. ($\eta_{rewr} : \{\rightarrow, \forall\}$) $\xi(\mu(r, s)) \triangleright s$
22. ($\eta_{rewr} - \exists$) $\mu(s, \xi(r)) \triangleright s$
23. $\sigma(\tau(r, s)) \triangleright \tau(\sigma(s), \sigma(r))$
24. $\sigma(\text{sub}_L(r, s)) \triangleright \text{sub}_R(\sigma(s), \sigma(r))$
25. $\sigma(\text{sub}_R(r, s)) \triangleright \text{sub}_L(\sigma(s), \sigma(r))$
26. $\sigma(\xi(r)) \triangleright \xi(\sigma(r))$
27. $\sigma(\xi(s, r)) \triangleright \xi(\sigma(s), \sigma(r))$
28. $\sigma(\mu(r)) \triangleright \mu(\sigma(r))$
29. $\sigma(\mu(s, r)) \triangleright \mu(\sigma(s), \sigma(r))$
30. $\sigma(\mu(r, u, v)) \triangleright \mu(\sigma(r), \sigma(u), \sigma(v))$
31. $\tau(r, \text{sub}_L(\rho, s)) \triangleright \text{sub}_L(r, s)$
32. $\tau(r, \text{sub}_R(s, \rho)) \triangleright \text{sub}_L(r, s)$
33. $\tau(\text{sub}_L(r, s), t) \triangleright \tau(r, \text{sub}_R(s, t))$
34. $\tau(\text{sub}_R(s, t), u) \triangleright \text{sub}_R(s, \tau(t, u))$
35. $\tau(\tau(t, r), s) \triangleright \tau(t, \tau(r, s))$

Theorem 2.10 (Termination property, [20]) *$LND_{EQ}\text{-TRS}$ is terminating.*

Theorem 2.11 (Confluence property, [20]) *$LND_{EQ}\text{-TRS}$ is confluent.*

3 Finale

The conception of the very first decision procedures for first-order sentences in the 1920's brought about the need for giving 'logical' citizenship to function symbols (e.g., Skolem functions). We have taken the view that a closer look at proof procedures for first-order sentences with equality brings about the need for introducing what we have called the "missing entity": (function) symbols for rewrites. This, we have argued, is appropriately done via the framework of labelled natural deduction which allows to formulate a proof theory for the "logical connective" of propositional equality. The basic idea is that when analysing an equality sentence into (i) proof conditions (*introduction*) and (ii) immediate consequences (*elimination*), it becomes clear that we need to bring in identifiers (i.e. function symbols)

for sequences of rewrites, and this is what we have claimed should be the missing entity in P. Martin-Löf’s equality types, both intensional and extensional. What we end up with is a formulation of what appears to be a middle ground solution to the ‘intensional’ vs. ‘extensional’ dichotomy which permeates most of the work on characterising propositional equality in natural deduction style.

Uniqueness of equality proofs.

Although the rewriting system is terminating and confluent, we have observed an interesting phenomenon here: there may be more than a normal proof of an equality statement. This is not a contradiction since the confluence property only says that the term for the equality reason can be brought to a unique normal form regardless of the order in which it is reduced. But there may be a different, yet normal, proof of the same equality statement. What appears to be a similar phenomenon has been identified in a paper by Hoffman and Streicher [9], which, although in a different context, indicates that there is a need to investigate further what is happening here.

References

- [1] P. H. G. Aczel. Frege Structures and the Notions of Proposition, Truth and Set, in J. Barwise, H.-J. Keisler, and K. Kunen (eds.), *The Kleene Symposium*, Vol. 101 of *Studies in Logic and The Foundations of Mathematics*, pp 31–59, North-Holland Publishing Co., Amsterdam, xx+425pp, 1980. Proceedings of the Symposium held in June 18–24, 1978, at Madison, Wisconsin, USA
- [2] P. H. G. Aczel. Term Declaration Logic and Generalised Composita, in *Sixth Annual IEEE Symposium on Logic in Computer Science (LICS’91)*, pp 22–30, IEEE Press, 1991. Proceedings of the Symposium held July 15–18 1991, in Amsterdam, The Netherlands
- [3] H.P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundation of Mathematics*. North-Holland. Amsterdam. New York. Oxford., revised edition, 1984.
- [4] Errett Bishop. *Foundations of Constructive Analysis*. McGraw-Hill series in Higher Mathematics. McGraw-Hill Book Company, New York, xiv+371pp., 1967.
- [5] Ph. Le Chenadec. On the logic of unification. *J. Symbolic Computation*, 8(1 and 2):141–199, July/August 1989.
- [6] D. M. Gabbay. *Labelled Deductive Systems, Volume I - Foundations*. Oxford University Press, 1996.
- [7] D. M. Gabbay and R. J. G. B. de Queiroz. Extending the Curry-Howard interpretation to linear, relevant and other resource logics. *The Journal of Symbolic Logic*, 57(4):1319–1365, December 1992.
- [8] J. R. Hindley and J. P. Seldin. *Lambda Calculus and Combinators. An Introduction*. Cambridge University Press, 2008.
- [9] M. Hofmann and T. Streicher. The groupoid model refutes uniqueness of identity proofs. In *Logic in Computer Science, 1994 (LICS ’94)*, pp. 208–212, 1994.
- [10] W. A. Howard. The formulae-as-types notion of construction. In J. R. Seldin and J.R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic Lambda Calculus and Formalism*. Academic Press, 1980. xxv+606pp.
- [11] G. Kreisel and W. Tait. Finite definability of number theoretic functions and parametric completeness of equational calculi. *Zeitschr. f. Math. Logik und Grundlagen d. Math.*, 7:28–38, 1961.
- [12] P. Martin-Löf. About Models for Intuitionistic Type Theories and the Notion of Definitional Equality, in S. Kanger (ed.), *Proceedings of the Third Scandinavian Logic Symposium*, Series *Studies in Logic and The Foundations of Mathematics*, pp 81–109, North-Holland, Amsterdam, 1975. Symposium held in 1973

- [13] P. Martin-Löf. An intuitionistic theory of types: predicative part, in H. E. Rose and J. C. Shepherdson (eds.), *Logic Colloquium '73*, Vol. 80 of *Studies in Logic and The Foundations of Mathematics*, pp 73–118, North-Holland, Amsterdam, viii+513pp, 1975. Proceedings of the Colloquium held in Bristol, UK, in 1973
- [14] P. Martin-Löf. Constructive Mathematics and Computer Programming, in L. J. Cohen, J. Los, H. Pfeiffer, and K.-P. Podewski (eds.), *Logic, Methodology and Philosophy of Science VI*, Series *Studies in Logic and The Foundations of Mathematics*, pp 153–175, North-Holland, Amsterdam, xiii+738pp, 1992. Proceedings of the International Congress held in Hannover, August 22–29 1979
- [15] P. Martin-Löf. *Intuitionistic Type Theory*. Series Studies in Proof Theory. Bibliopolis Naples, iv+91pp., 1984. Notes by Giovanni Sambì of a series of lectures given in Padova, June 1980.
- [16] J. C. Mitchell and A. Scedrov. Notes on scoping and relators. In E. Boerger et al., editor, *Computer Science Logic '92, Selected Papers*, pages 352–378. Springer LNCS 702, 1993. Available by anonymous ftp from host ftp.cis.upenn.edu and the file pub/papers/scedrov/rel.dvi.
- [17] B. Nordström, K. Petersson, and J. M. Smith. *Programming in Martin-Löf's Type Theory. An Introduction*, Vol. 7 of *The International Series of Monographs on Computer Science*, Clarendon Press, Oxford, x+221pp, 1990.
- [18] A. G. de Oliveira. Proof Transformations for Labelled Natural Deduction via Term Rewriting. (In Portuguese). Master's thesis, Depto. de Informática, Universidade Federal de Pernambuco, C.P. 7851, Recife, PE 50732-970, Brasil, April 1995.
- [19] A. G. de Oliveira and R. J. G. B. de Queiroz. Term Rewriting Systems with *Labelled Deductive Systems*. *Proceedings of Brazilian Symposium on Artificial Intelligence (SBIA'94)*, Lecture Notes in Artificial Intelligence, Springer, pages 59–72, 1994.
- [20] A. G. de Oliveira and R. J. G. B. de Queiroz. A Normalization Procedure for the Equational Fragment of Labelled Natural Deduction'. *Logic Journal of the Interest Group in Pure and Applied Logics* 7(2):173–215, 1999.
- [21] A. G. de Oliveira and R. J. G. B. de Queiroz. A New Basic Set of Proof Transformations. In *We Will Show Them! Essays in Honour of Dov Gabbay. Volume 2*, S. Artemov, H. Barringer, A. Garcez, L. Lamb and J. Woods (eds.), College Publications, London, ISBN 1904987125, October 2005, pp. 499–528.
- [22] R. J. G. B. de Queiroz. A proof-theoretic account of programming and the rôle of reduction rules. *Dialectica* 42(4):265–282, 1988.
- [23] R. J. G. B. de Queiroz. The mathematical language and its semantics: to show the consequences of a proposition is to give its meaning. In Weingartner, P. and Schurz, G., editors, *Reports of the Thirteenth International Wittgenstein Symposium 1988*, volume 18 of *Schriftenreihe der Wittgenstein-Gesellschaft*, Vienna, 1989, pp. 259–266, Hölder-Pichler-Tempsky. Symposium held in Kirchberg/Wechsel, Austria, August 14–21 1988.
- [24] R. J. G. B. de Queiroz. Meaning as grammar *plus* consequences. *Dialectica* 45(1):83–86, 1991.
- [25] R. J. G. B. de Queiroz. *Grundgesetze* alongside *Begriffsschrift* (abstract). In *Abstracts of Fifteenth International Wittgenstein Symposium*. pp. 15–16. Symposium held in Kirchberg/Wechsel, August 16–23 1992.
- [26] R. J. G. B. de Queiroz. Normalisation and Language-Games. *Dialectica* 48(2):83–125, 1994.
- [27] R. J. G. B. de Queiroz. Meaning, function, purpose, usefulness, *consequences* – interconnected concepts. *Logic Journal of the Interest Group in Pure and Applied Logics* 9(5):693–734, 2001.
- [28] R. J. G. B. de Queiroz. On reduction rules, meaning as use, and proof-theoretic semantics *Studia Logica* 90(2):211–247, November 2008.
- [29] R. J. G. B. de Queiroz and D. M. Gabbay. Equality in Labelled Deductive Systems and the functional interpretation of propositional equality. In P. Dekker and M. Stokhof, editors, *Proceedings of the 9th Amsterdam Colloquium*, pages 547–546, 1994.
- [30] R. J. G. B. de Queiroz and D. M. Gabbay. The functional interpretation of the existential quantifier. *Bulletin of the Interest Group in Pure and Applied Logics*, 3(2 and 3):243–290, 1995. Abstract in JSL 58(2):753–754, 1993. (Presented at *Logic Colloquium '91*, Uppsala, August 9–16.)
- [31] R. J. G. B. de Queiroz and D. M. Gabbay. The functional interpretation of modal necessity. In de Rijke, Maarten, editor, *Advances in Intensional Logic*, Applied Logic Series, Kluwer Academic Publishers, September 1997, pp. 61–91.
- [32] R. J. G. B. de Queiroz and D. M. Gabbay. Labelled Natural Deduction. In *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay's 50th Anniversary*, H.J. Ohlbach and U. Reyle (eds.), Kluwer Academic Publishers, June 1999, pp. 173–250.

- [33] R. J. G. B. de Queiroz, D. M. Gabbay, and A. G. de Oliveira. *The Functional Interpretation of Logical Deduction*, Imperial College Press, World Scientific, to appear.
- [34] R. J. G. B. de Queiroz and A. G. de Oliveira. Natural Deduction for Equality: The Missing Entity. In *Advances in Natural Deduction*, E. H. Haeusler, L. C. Pereira & V. de Paiva (eds.), a volume of the series *Trends in Logic*, Kluwer/Springer, 2010, to appear.
- [35] R. Statman. Herbrand’s Theorem and Gentzen’s notion of a direct proof. In J. Barwise, editor, *Handbook of Mathematical Logic*. North-Holland, Amsterdam, 1977.
- [36] R. Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic* 15:225–287, 1978.
- [37] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction. Vol. II*, Vol. 123 of *Studies in Logic and The Foundations of Mathematics*, North-Holland, Amsterdam, xvii+535pp, 1988.